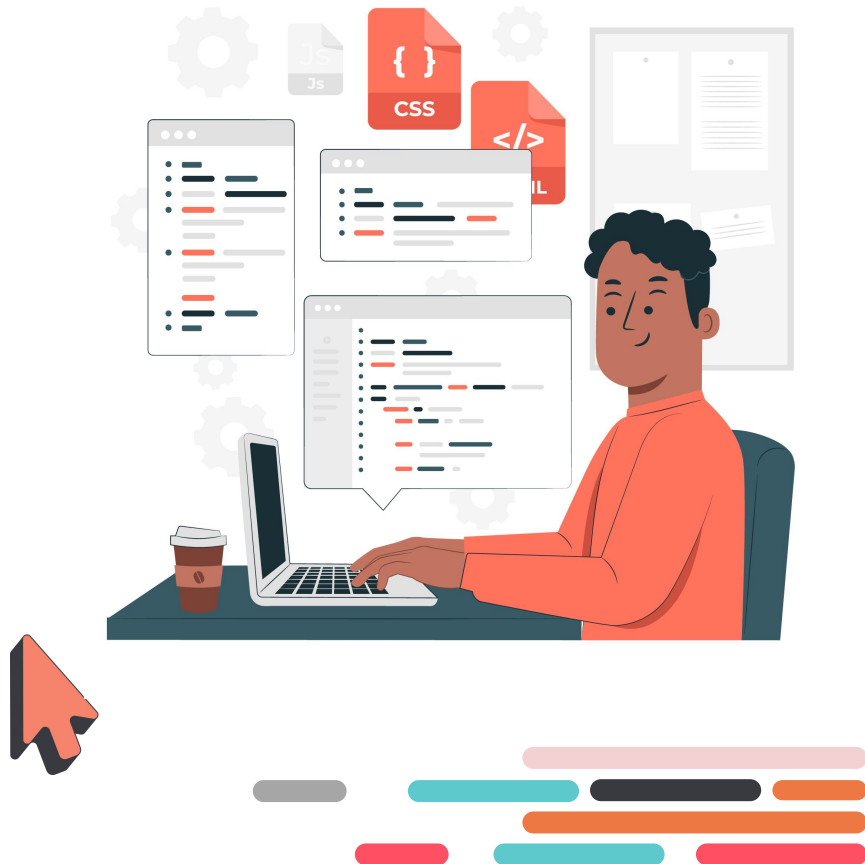




KEMNAKER



Pengenalan ReactJS, Components, Props & State Lifecycle





KEMNAKER

sanbercode



Outline

- Pengenalan Reactjs
- Instalasi Reactjs
- Components & Props
- State Life cycle





Pengenalan ReactJS





Apa itu ReactJS?

ReactJS atau **React** adalah **Javascript library** yang digunakan untuk membangun user interface (antar muka), jadi react itu adalah sebuah library bukan sebuah framework. Kalian bisa membaca tentang React adalah library disini. Reactjs sendiri di kembangkan oleh facebook (public release 29 May 2013).

React dibuat oleh Jordan Walke, seorang software engineer di Facebook, yang merilis prototipe awal React yang disebut “FaxJS “. React ini juga dapat digunakan sebagai basis dalam pengembangan Web App atau mobile. Namun, React hanya berkaitan dengan manajemen status dan rendering status itu ke DOM, jadi membuat aplikasi React biasanya memerlukan penggunaan library tambahan untuk perutean, serta fungsionalitas sisi klien tertentu.





Hal-hal Penting dalam Reactjs

berikut ini hal-hal yang penting dalam reactjs:

1. **Components**
2. **JSX**
3. **Virtual DOM**
4. **States & Lifecycle**





Components

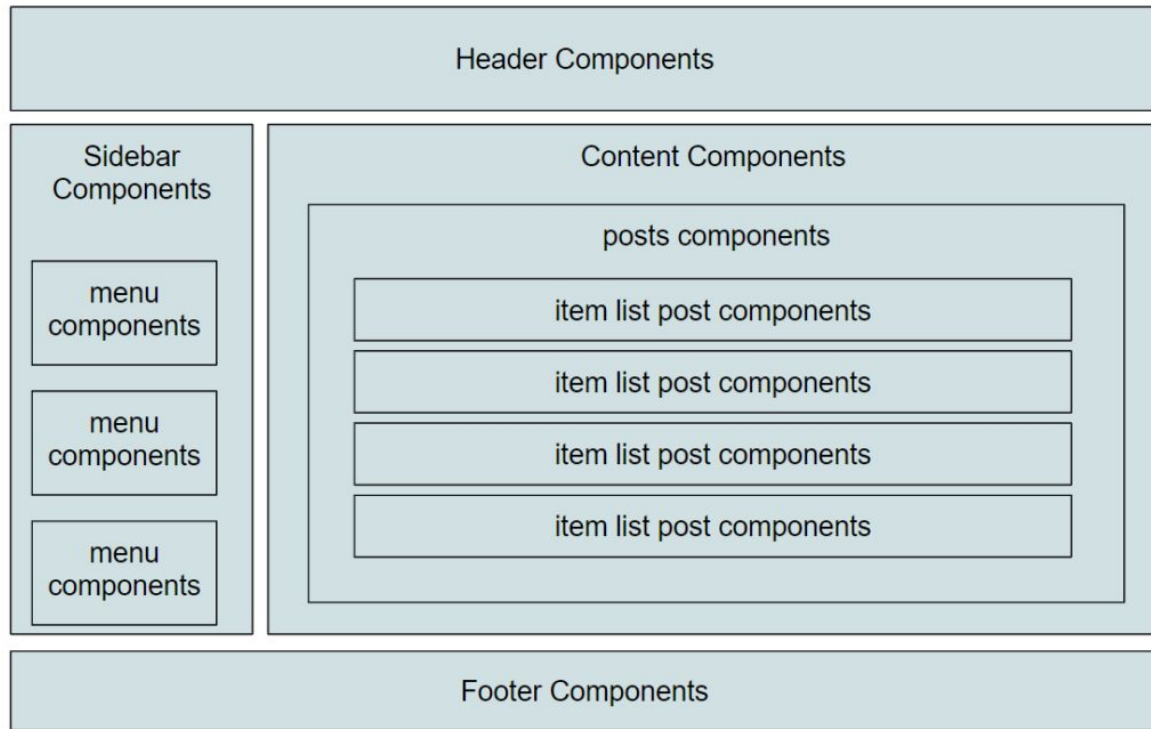
Components bisa di bilang hal yang sangat penting di reactjs. Components adalah semacam konsep bagaimana cara untuk membuat tiap bagian dari UI (tampilan) bisa bekerja secara mandiri dan terisolasi dari bagian-bagian yang lain.

Sebenarnya secara sadar atau tidak sadar bagi yang sering bekerja dengan kode HTML, selalu melakukan pemisahan bagian dari halaman-halaman misalnya **Header, Content, dan Footer**. Biasanya tujuannya agar kode tidak menumpuk di satu halaman sehingga dipisahkan ke file-file kecil yang lebih spesifik fungsinya dan kode lebih maintainable jika ada perbaikan.





Ilustrasi Components pada ReactJS





JSX

JSX merupakan ekstensi javascript yang membuat kita dapat menuliskan tag HTML di dalam javascript.

```
const element = <h1>Hello, world!</h1>;
```

pada reactjs ini kita akan JSX dan mungkin ada sedikit perbedaan di dalamnya, tetapi untuk yang sudah memahami html ini tidak akan terlihat sulit





KEMNAKER



Virtual DOM

Jika kita mulai mendalami React, akan ada pertanyaan apa itu virtual DOM? Virtual DOM adalah representasi DOM secara virtual.

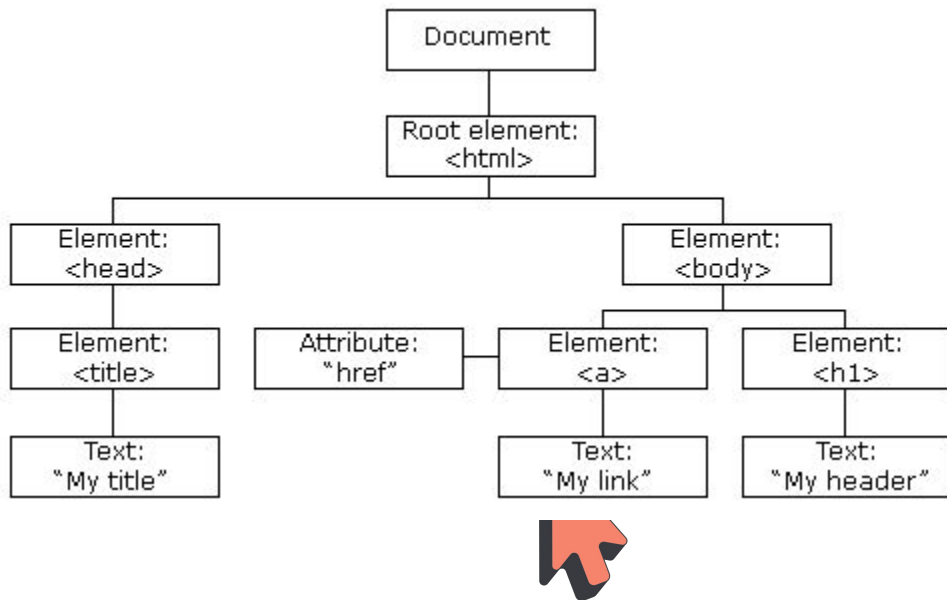
JavaScript itu sebenarnya berjalan dengan cepat, yang menyebabkan JavaScript terasa lambat adalah ketika JavaScript mengolah DOM.

Apa itu dom ? DOM adalah representasi dari UI pada aplikasi kita. Secara singkat bentuk DOM sendiri adalah struktur data tree, karena itu jika adanya perubahan atau update ke dom akan cepat. Namun, setelah adanya perubahan pada dom, element yang dirubah dari mulai parent hingga child nya akan dirender ulang untuk mengupdate UI di aplikasi kita. Proses re-rendering atau re-painting UI ini lah yang membuat lambat





Ilustrasi DOM





State

State adalah asal dari suatu data. Komponen pada react tentunya membutuhkan data (tidak semua namun rata-rata membutuhkan data). Data tersebut dapat berasal dari mana saja. State adalah salah satu sumber data tersebut. Selain state data dari komponen juga dapat berasal dari props (bukan property).





Lifecycle

Terdapat tiga Lifecycle dalam sebuah komponen dalam react yaitu :

1. Inisialisasi / mounting (ketika komponen dibuat/ditambahkan pertama kali pada DOM)
2. Update / rerender (ketika terdapat perubahan state/prop yang mengakibatkan perubahan pada DOM)
3. Unmounting (ketika komponen akan dihapus dari DOM)

Pada setiap lifecycle tersebut, komponen react akan mengeksekusi method/fungsi yang berbeda yang kita sebut sebagai lifecycle methods. Beberapa method mempunyai prefix will dan did yang menunjukkan kapan method tersebut akan dieksekusi.





Instalasi ReactJS





KEMNAKER



Instalasi

sebelum ke instalasi dan setup ada beberapa requirements atau prasyaratnya yaitu:

1. **nodeJS (versi LTS boleh ataupun yang terbaru) tapi disini saya menyarankan agar menginstal versi yang LTS juga cukup**
2. **text editor (visual studio code, sublime, atom dll)**
3. **terminal (bisa cmd, git bash atau terminal di text editor)**





Step Instalasi

Berikut Step Instalasi:

1. Instalasi bisa menggunakan `npm create vite@latest my-app -- --template react`
2. setelah selesai change directory ke folder my-app
3. silahkan edit file app.jsx untuk merubah kode reactnya dan lihat hasilnya





KEMNAKER



Sesi Tanya Jawab





Components





KEMNAKER



Apa itu Component React?

Seperti yang sudah disampaikan sebelumnya **Component** bisa dibilang hal yang sangat penting di react.js. Components adalah semacam konsep bagaimana cara untuk membuat tiap bagian dari UI (tampilan) bisa bekerja secara mandiri dan terisolasi dari bagian-bagian yang lain.

Component React dari pembuatannya terbagi dua yaitu **class component** dan **function component**






Contoh Component

Class Component:

```
class Welcome extends React.Component {  
  render() {  
    return <h1>Hello, {this.props.name}</h1>;  
  }  
}
```

Function Component:

```
function Welcome(props) {  
  return <h1>Hello, {props.name}</h1>;  
}
```





KEMNAKER



Apa itu Props?

Sedangkan Props, merupakan properti-properti dari component yang dapat kita gunakan untuk menampilkan data, biasanya digunakan untuk inisialisasi data dengan kata lain bahwa props adalah “model” data yang memiliki fungsi ganda yaitu selain untuk menyimpan data, Props juga berfungsi untuk “melempar data” antar components view jika ada perubahan data di state.





Contoh Penggunaan Props

```
function Welcome(props) {  
  return <h1>Hello, {props.name}</h1>;  
}  
  
function App() {  
  return (  
    <div>  
      <Welcome name="Sarah" />  
      <Welcome name="Michael" />  
      <Welcome name="John" />  
    </div>  
  );  
}
```





Import dan Export

Import dan Export sendiri tujuannnya agar syntax yang kita tuliskan tidak menumpuk didalam 1 file, kita bisa membedakan syntax (componentnya) ke file yang lain



Contoh Import Export

misal ada file **Welcome.js**:

```
function Welcome(props){  
  return <h1>Hello, {props.name}</h1>;  
}  
export default Welcome
```

lalu Component **Welcome** tersebut di import dan digunakan pada component **App**

```
import Welcome from './Welcome';  
function App() {  
  return (  
    <div>  
      <Welcome name="Sarah" />  
      <Welcome name="Michael" />  
      <Welcome name="John" />  
    </div>  
  );  
}  
export default App
```





State & Components Lifecycle





Apa itu State?

Sebelum kita masuk pada penjelasan component lifecycle, Kita harus mengetahui dulu tentang state.

State merupakan data yang tersimpan dalam sebuah component. State bersifat private dan hanya relevan terhadap component itu sendiri. Berbeda dengan props yang valuenya dilempar dari component lain, state justru dapat menyimpan dan mengubah datanya sendiri dari dalam component.





Ilustrasi Lifecycle pada class component

Initialization

setup props and state

Mounting

componentWillMount

render

componentDidMount

Updation

props

componentWillReceiveProps

shouldComponentUpdate

componentWillUpdate

render

componentDidUpdate

states

shouldComponentUpdate

componentWillUpdate

render

componentDidUpdate

Unmounting

componentWillUnmount

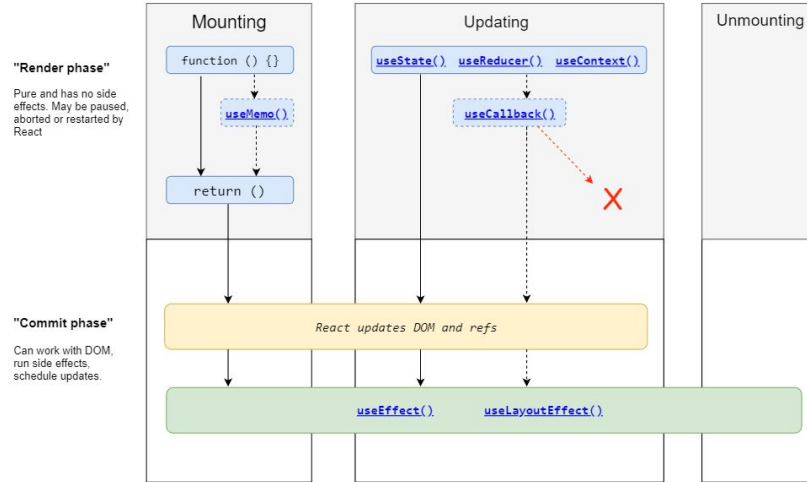




Ilustrasi Lifecycle pada functional Component



React hooks lifecycle - function component
version 16.8.x



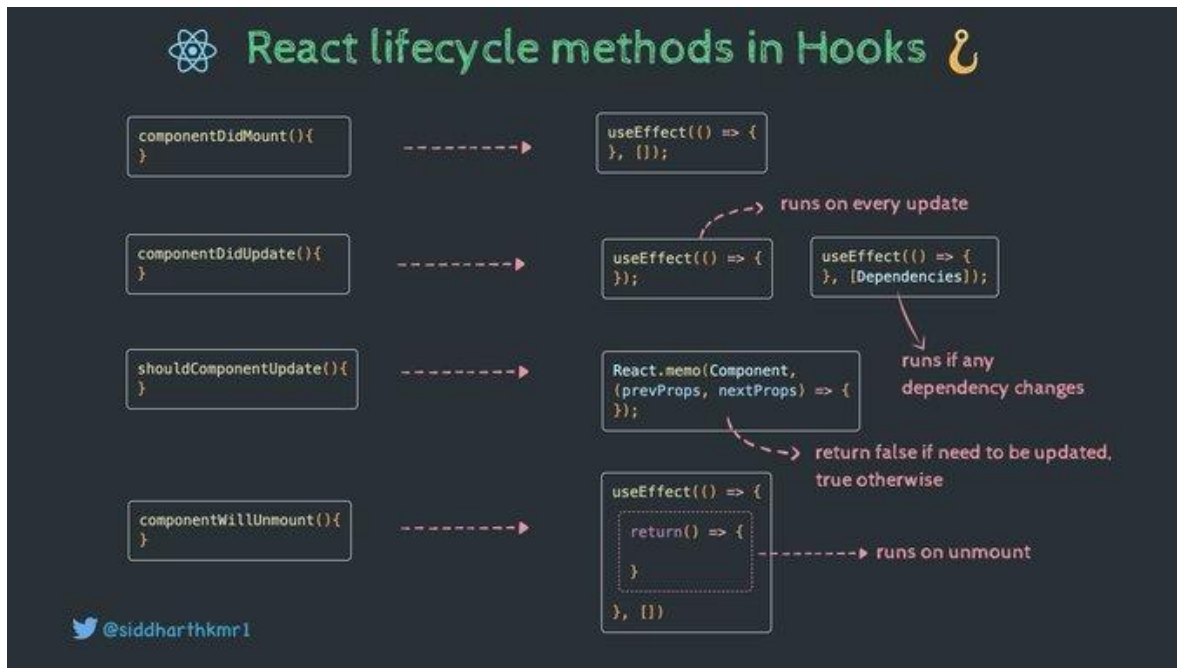
Made with <3 by Gal Margalit. feel free to contribute and share  [wavez/react-hooks-lifecycle](https://github.com/wavez/react-hooks-lifecycle)



KEMNAKER



Perbandingan Hooks dan lifecycle class component





KEMNAKER

sanbercode



Kesimpulan

- Pengenalan Reactjs
- Instalasi Reactjs
- Components & Props
- State Life cycle





Terima Kasih

